

5 METHOD AND SYSTEM FOR
OBSCURING USER ACCESS PATTERNS
USING A BUFFER MEMORY

FIELD OF THE INVENTION

The present invention relates generally to computer networks and, more particularly, to a method and system that employs a cache memory, also known as a buffer memory, in combination with a reference editing function to obscure user access patterns. 10 The present invention is particularly applicable in open computer networks, such as the Internet.

BACKGROUND OF THE INVENTION

15 As used herein, the term "computer" includes any device or machine capable of accepting data, applying prescribed processes to the data, and supplying the results of the processes. By way of example, but not limitation, the term "computer" includes mainframe computers, servers, personal computers, laptops, personal digital assistants, portable phones, cell phones and calculators. The term "communications network" is also meant in a broad 20 sense, and may include any suitable technology for information transmission, including electrical, electromagnetic and optical technologies. Such a communications network may link computers, e.g., a LAN or WAN. Although the invention is described with particular reference to an open network, such as the Internet, it may also be used in other networks, internets and intranets.

25 The Internet, or World Wide Web, continues to increase in importance as a place for business, offering a wide variety of information and services to potential customers. However, shopping, browsing and other information-sharing activities on the Internet expose users to unwanted collection of their private and personal information, from which their identities, activities, behaviors and preferences can be ascertained. For 30 example, without a user's permission, web marketers and merchants often gather "click data" that details every web-site a user visits with his or her browser. Underlying communications protocols and systems may provide additional private and/or personal information. This data is then used to create demographic profiles linked with the user's identity, including his or her name, postal address and e-mail address, gender, age, and other 35 personal information. This information is routinely bought and sold among parties who link and merge the information with other transaction data from other sources (i.e., "data

mining") offered for sale by third parties and vendors to create a sophisticated and detailed behavior profile of users, in order to target those users for advertising. This unwarranted level of intrusion into the private information of a user, often unknown to the user, is perceived as a fundamental threat to personal freedoms, creating an outcry among a number
5 of privacy groups and a potential impediment to the growth of e-commerce. U.S. Patent Application Serial No. 09/360,812, to the present inventor, which discusses these privacy concerns and discloses a system and method for anonymous Internet transactions, is hereby incorporated by reference.

10 SUMMARY OF THE INVENTION

In some computer network applications it is desirable to obscure user access patterns. For example, some users may want to hide their interest in a referenced object (i.e., a site on the World-Wide Web or a specific item of information located on such a site), and/or prevent their identities from being discovered and tracked by others. The present
15 invention uses a cache memory, also known as a buffer memory, to obscure patterns in the reference behavior of a user seeking one or more objects accessible through the cache memory. An application using the World-Wide Web infrastructure is used to illustrate the concept.

In a preferred embodiment, the present invention provides a method for
20 obscuring user requests for information in a computer network. A user computer request for information, aimed at another network member, is routed to a first cache memory. If the first cache memory contains the requested information, the requested information is returned in response to the user request without releasing the user request to the network member. If, however, the first cache memory does not contain the requested information,
25 user identity information contained in the request is edited, resulting in an edited request with obscured identity information. The edited request is then released to the network member, the user computer receives the requested information from the network member, and a copy of the requested information is stored in the first cache memory. In this method, user requests for information that can be satisfied by information stored in the cache
30 memory are not revealed to other network members, and user requests that cannot be satisfied by the cache memory are obscured by editing prior to release to other network members. The communications network may be the Internet, and the user request for information may be a Uniform Resource Locator (URL) reference string.

The edited request may be routed to a second cache memory prior to
35 releasing the edited request to the network member and, if the second cache memory

DEPARTMENT OF COMMERCE

contains the requested information, the requested information is returned to the user computer in response to the edited request without releasing the edited request to the network member. In addition, if the second cache memory does not contain the requested information, the user identity information contained in the request is further edited prior to

5 releasing the edited request to the network member.

The first cache memory may be resident on the user computer, or it may be resident on a computer remote from the user computer. Similarly, the editing may be performed by the user computer, or it may be performed by a computer remote from the user computer.

10 In another embodiment, the present invention is a software program implemented in a computer system for obscuring user requests for information on a computer network. The software program configures the computer system to (1) route a user computer request for information, aimed at another network member, to a first cache memory; (2) if the first cache memory contains the requested information, return the

15 requested information in response to the user request without releasing the user request to the network member; (3) if the first cache memory does not contain the requested information, edit user identity information contained in the request, resulting in an edited request with obscured identity information; (4) release the edited request to the network member; (5) receive the requested information from the network member; and (6) store a

20 copy of the requested information in the first cache memory such that user requests for information that can be satisfied by information stored in the cache memory are not revealed to other network members, and user requests that cannot be satisfied by the cache memory are obscured by editing prior to release to other network members. The computer network may be the Internet, and the user computer request for information may be a Uniform

25 Resource Locator (URL) reference string. The network member may be a server computer. The cache memory may be resident on the user computer or on a computer remote from the user computer.

 The software program of the present invention may further configure the computer system to route the edited request to a second cache memory prior to releasing the

30 edited request to the network member, and, if the second cache memory contains the requested information, return the requested information in response to the edited request without releasing the edited request to the network member. If the second cache memory does not contain the requested information, the software program may further configure the computer system to further edit the user identity information contained in the edited request

35 prior to releasing the edited request to the network member.

In still another embodiment, the invention is a system for obscuring user requests for information in a computer network comprising a user computer having a processor, a server computer containing information sought by the user computer, a first cache memory capable of storing information, and a reference editing function capable of 5 editing user identity information contained in a request for information. A request for information from the user computer, aimed at the server computer, is routed by the processor to the first cache memory. If the first cache memory contains the requested information, the requested information is returned in response to the user request without releasing the user request to the server computer. However, if the first cache memory does 10 not contain the requested information, the first reference editing function edits user identity information contained in the request prior to releasing the request to the server computer, and upon receipt of the requested information from the server computer, the first cache memory stores a copy of the requested information. Thus, user requests for information that can be satisfied by information stored in the first cache memory are not revealed to other 15 network members, and user requests that cannot be satisfied by the first cache memory are obscured by the first reference editing function prior to release to the server computer. The computer network may be the Internet, and the user computer request for information may be a Uniform Resource Locator (URL) reference string. The first cache memory may be resident on the user computer or on a computer remote from the user computer. The first 20 reference editing function may also be resident on the user computer or on a computer remote from the user computer.

In one specific embodiment, the system further comprises a second cache memory wherein the edited request for information is routed to the second cache memory prior to release to the server computer, and if the second cache memory contains the 25 requested information, the requested information is returned in response to the edited request without releasing the edited request to the server computer. The system may also comprise a second reference editing function, wherein if the second cache memory does not contain the requested information, the second reference editing function further edits the user identity information contained in the edited request prior to releasing the edited request 30 to the server computer.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated more fully from the following detailed description, taken in conjunction with the drawings in which:

FIG. 1 is a table illustrating buffer cache behavior with a memory and a reference string;

FIG. 2 is flowchart illustrating the steps in a preferred embodiment of the method of the present invention;

5 FIG. 3 is a block diagram illustrating the operation of a proxy cache in a computer network.

FIGS. 3A through 3D are block diagrams illustrating the operation of one or more proxy cache memories in combination with a reference editing function in a computer network.

10

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Ideally, computers would be equipped with infinitely large storage systems, with zero cost and zero access time, which never fail. While this is not possible, engineers have, over time, developed various techniques for approximating these desiderata. In 15 particular, the use of a small amount of "buffer" cache memory, consisting of a small portion of fast, expensive memory, in conjunction with a larger, cheaper, slower memory, has proven effective. Buffer caches are widely used in computer systems because computer programs exhibit "locality of reference" properties such that references in the program to memory locations tend to access repeatedly the same locations in memory. This makes it 20 feasible to reduce system costs and improve average access time by storing the computer program in the slow memory and transferring referenced items from the slow memory to the fast memory, where they can be used subsequently. When references, arranged in a so-called "reference string," are made against the combined memories, access time is improved whenever the reference string repeatedly accesses the same locations, so that the 25 references can be satisfied with the copy in fast memory.

Because propagation and queuing delays strongly affect the responsiveness of networks, network systems such as the World Wide Web (WWW) attempt to use the buffer principle by interposing "proxy caches" in the network. Examples include Harvest and Squid.

30 In the World Wide Web context, the references discussed above are names – in the form of Universal Resource Locators (URLs). URLs are presented to the WWW infrastructure (web servers) that interpret strings such as "http://www.iPrivacy.com" to gain access to data that is eventually sent to the user's display by means of a software "Web browser" such as Netscape NAVIGATOR™ or Microsoft's INTERNET EXPLORER™.

As discussed above, a pervasive problem in the use of the Internet technology is the lack of privacy. Among the many organizations describing the extent and technologies for identity and behavior capture on the Internet are Junkbusters, Inc. (<http://www.junkbusters.org>). The reasons for this loss of privacy are manifold.

5 First, the basic Internet transport service, IP, provides a source and destination address in each data packet. The source address of the packet provides a means of identifying the sender.

10 Second, the technology used to access data over the Internet, the "web browser" (exemplified by Netscape and Microsoft Internet Explorer), provides many mechanisms with which users can be identified (e-mail addresses, login names, organization names, or "cookies" deposited on the user's machines).

15 Third, the mechanisms available have been exploited aggressively by organizations such as commercial enterprises which can derive advantage from greater knowledge of behavior. Examples of these advantages include better-directed advertising.

15 Finally, this data can be combined with other data (address, telephone, demographic) in data-mining exercises which thoroughly link behavior and identity.

This loss of privacy has led to the development of technologies providing various levels of anonymity when the Internet is being used. These technologies perform one or more of the following functions:

- 20 1. IP address rewriting (so the IP source address is lost)
2. Cookie deletion
3. Pseudonym provision / anonymous e-mail
4. HTML editing

As with Web caches, these technologies are typically implemented as Web "proxies." An 25 example of an anonymizing client proxy is the Internet Junkbuster proxy, found at <http://www.junkbusters.com>. An example of an anonymizing server proxy is the Lucent Personal Web Assistant, found at <http://www.proxymate.com>. The advantage of a server proxy is that with a location distinct from the client, it automatically rewrites IP addresses since the proxy's IP address is distinct from the client machine's.

30 It is an object of the present invention to combine a proxy cache with an anonymizing proxy to increase anonymity. This is accomplished by obscuring patterns in user references to the Web. Since the caching function and the anonymizing function (i.e., reference editing function) do not interfere with each other, the caching function can occur before or after the anonymizing function with the same effect. Further, multiple caching 35 proxies or multiple anonymizing proxies can be combined for further anonymity.

Reference is now made to FIG. 1, which is a table illustrating buffer cache behavior with a memory and reference string. Column 100 simply numbers successive reference requests; column 101 shows successive user reference requests (i.e., a reference string); column 102 depicts which reference requests can be satisfied by the contents of a simple four-element buffer memory; and column 104 shows the reference requests that cannot be satisfied by the current contents of the buffer memory and therefore are made to a large memory. For illustration purposes, the large memory is presumed to have 16 addresses, numbered 1 through 16 at which information is stored and the reference string consists of numerical indices to the addresses of the large memory. The contents of the buffer memory are the information found at the memory addresses specified by the reference requests listed in column 102. It should be understood that reference requests satisfied by accessing the large memory (column 104) are susceptible to tracking by an entity monitoring the large memory, while requests that are satisfied by the contents of the buffer memory (column 102) would be either significantly less susceptible or insusceptible to such tracking.

The table demonstrates how successive accesses to the large memory, column 104, are obscured when the buffer cache, column 102, successfully delivers the data requested. These events are marked by the word "Obscured" in column 104, which indicates that no reference request was made to the large memory.

At the time of the first request, the four element buffer, column 102 is empty. After each of the first four requests, a copy of the information stored at the address specified by the reference request is stored in one of the available spaces in the buffer memory. Thus, after the first four requests, the buffer is full. It should be understood that new reference requests will replace previously stored information associated with previous reference requests in the four element buffer when the information specified by new reference request is retrieved from the large memory. In this sense, the contents of the buffer is not fixed.

As shown in highlighted rows 106, 108, 110, the references to indices 05, 09 and 03, respectively, have been "obscured" because the buffer has provided the requested information. As the cache is more successful (i.e., as its "hit rate" increases), more and more references are obscured, and further, the inter-reference time varies with the success of the buffer cache. The effect of the cache on the reference string, column 104, is to obscure (i.e., prevent or inhibit others from tracking) a user's pattern of reference request (i.e., pattern of reference), column 101. Thus, as shown in FIG. 1, any entity tracking the references to the large memory will have less information concerning the user's access

patterns, because the tracking entity will not be able to record the user's request for indices 05, 09 and 03 (shown in rows 106, 108, and 110).

Reference is now made to FIG. 2, which is a flowchart illustrating the steps in a preferred embodiment of the method of the present invention. In step 150, a user computer requests information from another network member. In a preferred embodiment, the network member is a server computer. In step 155, the user request is routed to a proxy cache memory. This cache memory may be resident at the user computer or located remote from the user computer. In step 160, a test is made to determine if the cache memory has the requested information. If the cache has the requested information, the requested information is returned to the user computer at step 180. The process ends at step 185, and the system awaits the next request for information.

Returning to step 160, if the cache memory does not have the requested information, the request is edited in step 165 to obscure user identity information. The request is then released to the network in step 170, but due to the edited identity information, anyone attempting to trace the request will have difficulty linking it to the user computer. Assuming the request can be satisfied by a network member, in step 175 the requested information is received (either by the user computer or some proxy server) and a copy is stored in the cache memory. In step 180, the requested information is returned to the user computer. In step 185, the process ends, and the system awaits the next information request.

As illustrated in FIG. 3, in one embodiment the proxy cache 200 is placed between the browser (running on user computer 202) and the server 204. The cache examines URLs to see if the requested data can be obtained from its storage rather than by accessing the server. As in the traditional buffer cache example, this proxy cache can both reduce delays (if the proxy cache is "closer" to the client than the server, e.g., on a LAN) and traffic (since no network traffic to the server is needed).

In addition, a reference editing function is advantageously combined with the proxy cache so that the references generated by the user requests, when passed through the buffering/caching system, are obscured.

Four such combinations are illustrated in FIGS. 3A-3D. In FIG. 3A, a reference request (i.e., a URL reference string) from a user computer 202 is routed to a proxy cache 200. The proxy cache 200 may be resident at the user computer, or it may be located remote from the user computer and connected to the user computer via a communications network, such as the World Wide Web, or Internet. The cache 200 examines the URL to see if the requested data can be obtained from its storage rather than

by accessing the server. As in the traditional buffer cache example, this proxy cache can both reduce delays (if the proxy cache is "closer" to the client than the server, e.g., on a LAN) and traffic (since no network traffic to the server is needed). If the cache can satisfy the request, the information is returned to the user computer 202. If, however, the cache

5 200 does not contain the requested information, a reference editing function 206, edits the URL reference string to obscure the true identity of the user computer. The edited request is then released to the targeted server computer 204. In FIG. 3A, a communications network 210 connects the user computer 202 and the server computer 204. As with the proxy cache, the reference editing function 206 may be resident at the user computer 202 or located

10 remote from the user computer and connected to the user computer via a communications network. In addition, as shown in FIG. 3B, the order of the proxy cache 200 and reference editing function 206 may be reversed, resulting in the reference string being edited by the reference editing function 206 before it is released to the proxy cache 200.

FIGS. 3C and 3D illustrate systems with multiple proxy caches 200, 201.

15 The second proxy cache 201 may be resident at the user computer 202 (as shown in FIG. 3C) or remotely located from the user computer and connected to the user computer via communications network 210 (as shown in FIG. 3D). In addition, multiple reference editing functions may also be implemented, either resident at the user computer or remotely located.

20 While the present invention has been described with reference to the preferred embodiments, those skilled in the art will recognize that numerous variations and modifications may be made without departing from the scope of the present invention. Accordingly, it should be clearly understood that the embodiments of the invention described above are not intended as limitations on the scope of the invention, which is

25 defined only by the following claims.